OIPE
JUL 0 1 2005
PATENT & TRADEMARK OFFICE



2

**DATASTREAM**

**Request Packet from VT Server to host system**
**005000000000000000543762000331110000011110000000010000Y**

**Response Packet from host system to VT Server**
**005800000000000000543762033111000001111000+000075000+0000725**

Fig. 1

OIPE JC185 JUL 0 1 2005 PATENT & TRADEMARK OFFICE

APPLICATION PROGRAMMING INTERFACE

REQUEST PACKET

| FIELD DESCRIPTION | FORMAT | CONTENT |
|---|---|---|
| TRANSACTION CODE | 3N | '033' |
| CREDIT UNION ACCESS CODE | 3N | CODE ASSOCIATED WITH EACH CREDIT UNION ASSIGNED BY HOST |
| MEMBER NUMBER TO WITHDRAW FUNDS FROM | 9N | (ENTERED BY CALLER) |
| ACCOUNT SUFFIX TO WITHDRAW FUNDS FROM | 3N | (ENTERED BY CALLER) |
| TRANSFER AMOUNT | 9N | (TWO DECIMAL POSITIONED ASSUMED) |
| POST INDICATOR | 1A | N - PRELIMINARY EDIT<br>DO NOT UPDATE FILES |
| | | Y - MEMBER HAS CONFIRMED THEY WANT TO POST TRANSACTION UPDATE FILES |

RESPONSE PACKET

| FIELD DESCRIPTION | FORMAT | CONTENT |
|---|---|---|
| TRANSACTION CODE | 3N | '033' |
| CREDIT UNION ACCESS CODE | 3N | |
| MEMBER NUMBER | 9N | |
| HOST RESPONSE CODE | 3N | 000 - POSITIVE RESPONSE, CONTINUE SCRIPT |
| | | 210-214 - READ ERROR, REPEAT MENU |
| | | 220-221 - READ ERROR, REPEAT MENU |
| SIGN FIELD | 1A | + OR -, NEGATIVE OR POSITIVE BALANCE |
| CURRENT BALANCE OF WITHDRAWAL FROM ACCOUNT (BEFORE TRANSFER) | 9N | N - PRELIMINARY EDIT<br>DO NOT UPDATE FILES |
| SIGN FIELD | 1A | + OR -, NEGATIVE OR POSITIVE BALANCE |
| AVAILABLE BALANCE OF WITHDRAWAL FROM ACCOUNT (BEFORE TRANSFER) | 9N | |

FIG. 2

OIPE JC185 JUL 0 1 2005 PATENT & TRADEMARK OFFICE



MAPPING DOCUMENT

REQUEST PACKET

| API FIELD DESCRIPTION | VT SERVER FIELD DEFINITIONS |
|---|---|
| TRANSACTION CODE | HARD CODE - SET TO '033' |
| CREDIT UNION ACCESS CODE | CODE RETRIEVED FROM DATABASE CONFIGURATION FOR EACH PARTICULAR CREDIT UNION |
| MEMBER NUMBER TO WITHDRAW FUNDS FROM | FROM INTERNAL FIELD TRANSACACCNTFROM BEFORE FIELD SEPARATOR |
| ACCOUNT SUFFIX TO WITHDRAW FUNDS FROM | FROM INTERNAL FIELD TRANSACACCNTFROM AFTER FIELD SEPARATOR |
| TRANSFER AMOUNT | FROM INTERNAL FIELD TRANSACAMOUNT. INTERNAL FIELD INCLUDES DECIMAL POINT. EXTERNAL FIELD DOES NOT. REMOVE DECIMAL POINT BEFORE SENDING. MAXIMUM AMOUNT IS 9,999,999.99 |
| POST INDICATOR | SET BASED ON TRANSACPOSTMODE. IF 0, SET TO N. IF 1, SET TO Y. |

RESPONSE PACKET

| FIELD DESCRIPTION | CONSTANT |
|---|---|
| TRANSACTION CODE | FIELD IS ECHOED, NOT USED IN RESPONSE |
| CREDIT UNION ACCESS CODE | FIELD IS ECHOED, NOT USED IN RESPONSE |
| MEMBER NUMBER | FIELD IS ECHOED, NOT USED IN RESPONSE |
| HOST RESPONSE CODE | EXTERNAL SYSTEM'S RESPONSE CODE. MAP TO THE VT SERVER RESPONSE CODE BASED ON THE CONFIGURATION TABLES. |
| SIGN FIELD | POSITIVE/NEGATIVE SIGN INDICATOR FOR FIELD THAT FOLLOWS. USE TO MAP APPROPRIATELY. |
| CURRENT BALANCE OF WITHDRAWAL FROM ACCOUNT (BEFORE TRANSFER) | MAP TO THE LEDGER BALANCE FIELD OF THE INTERNAL MESSAGE. QBT BALANCE INDICATOR FOR LEDGER IS 1. |
| SIGN FIELD | POSITIVE/NEGATIVE SIGN INDICATOR FOR FIELD THAT FOLLOWS. USE TO MAP APPROPRIATELY. |
| AVAILABLE BALANCE OF WITHDRAWAL FROM ACCOUNT (BEFORE TRANSFER) | MAP TO THE LEDGER BALANCE FIELD OF THE INTERNAL MESSAGE. QBT BALANCE INDICATOR FOR LEDGER IS 2. |

**PRIOR ART**

**FIG. 3**

OIPE
JUL 0 1 2005
PATENT & TRADEMARK OFFICE



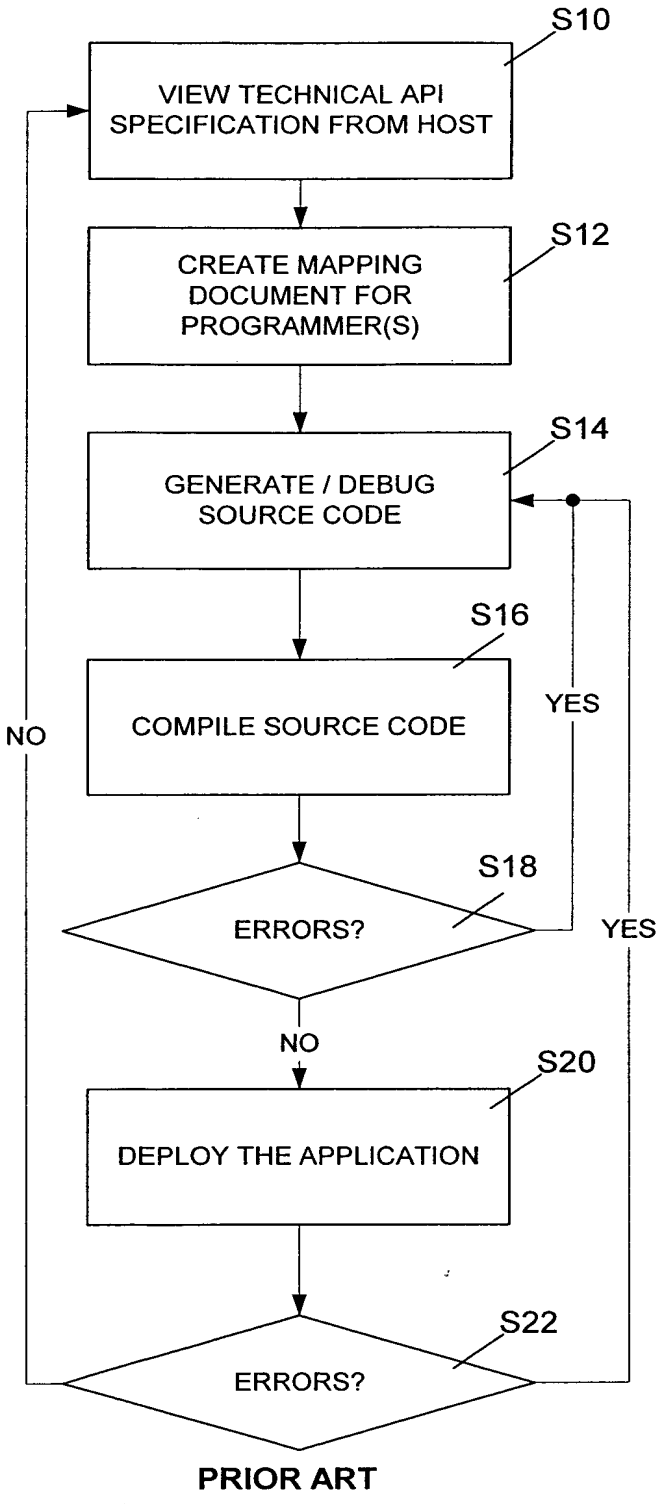


Fig. 4

OIPE
PATENT & TRADEMARK OFFICE
JUL 01 2005
JC18

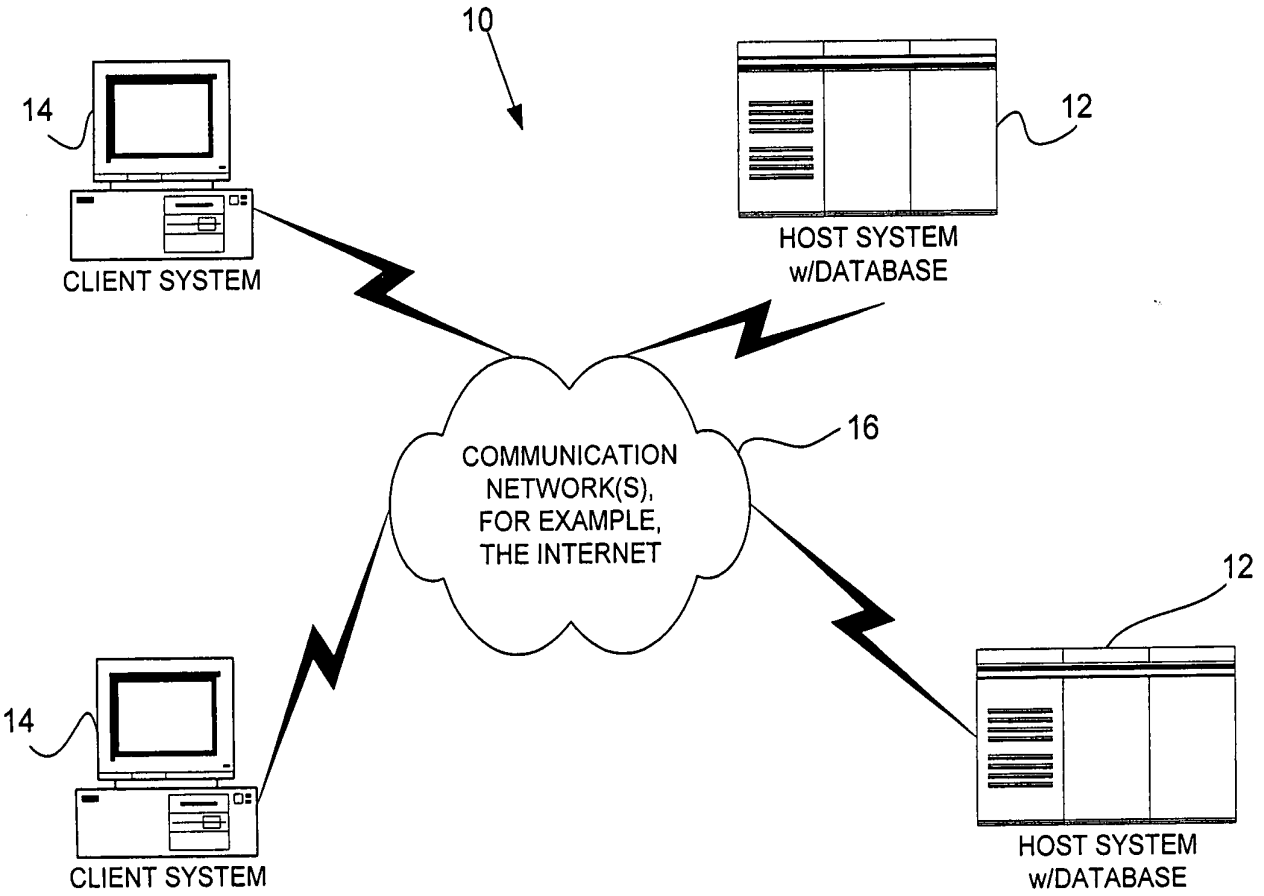


Fig. 5

OIPE
JUL 0 1 2005
PATENT & TRADEMARK OFFICE



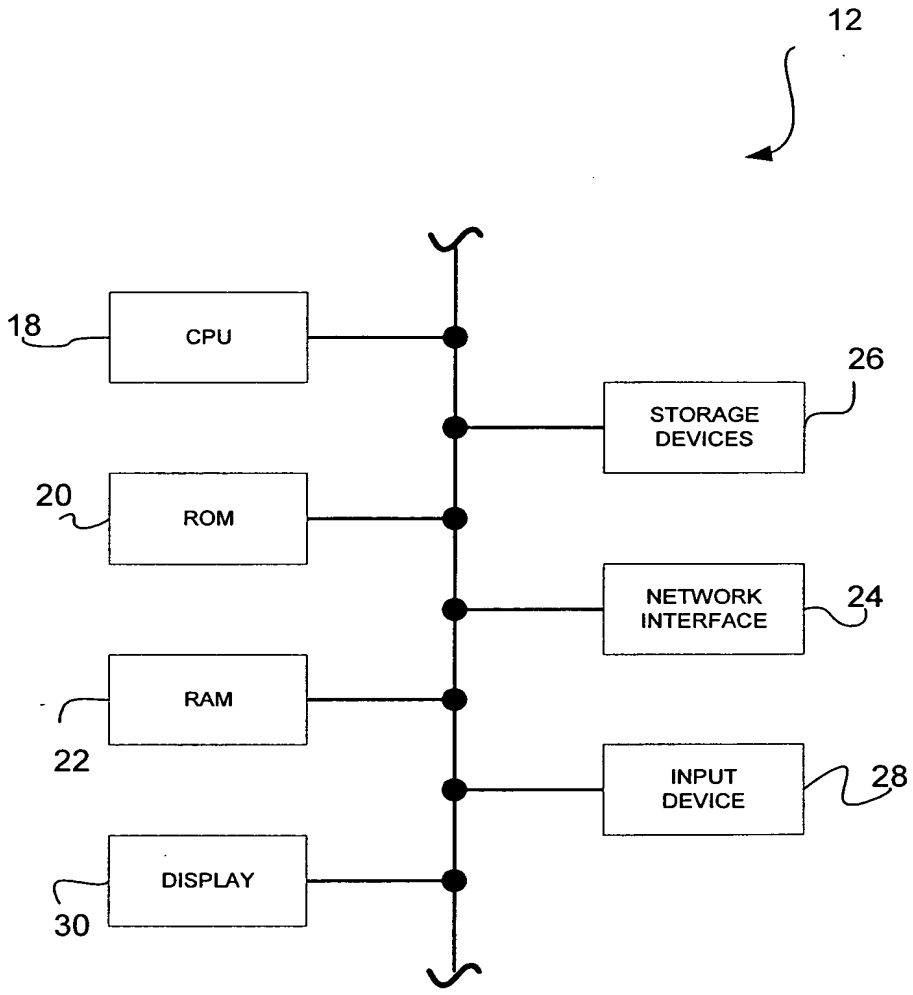


Fig. 6

OIPE
JUL 0 1 2005
PATENT & TRADEMARK OFFICE
JC105



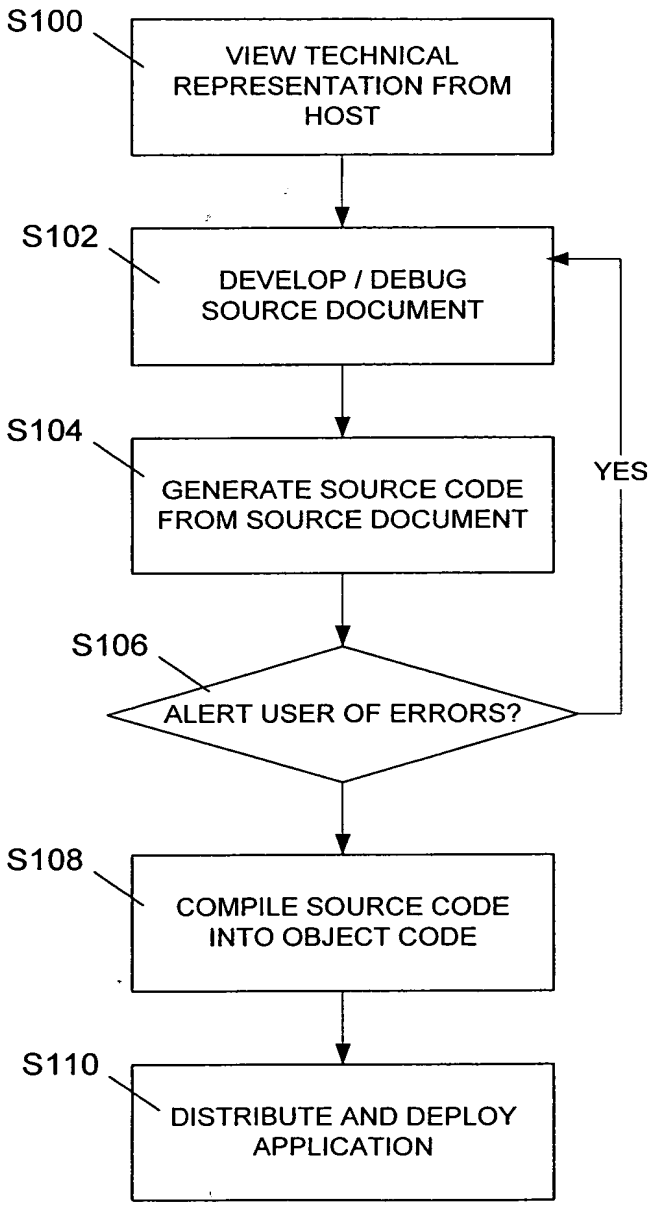


Fig. 7

PATENT & TRADEMARK OFFICE JUL 01 2005 JC196 O I P E

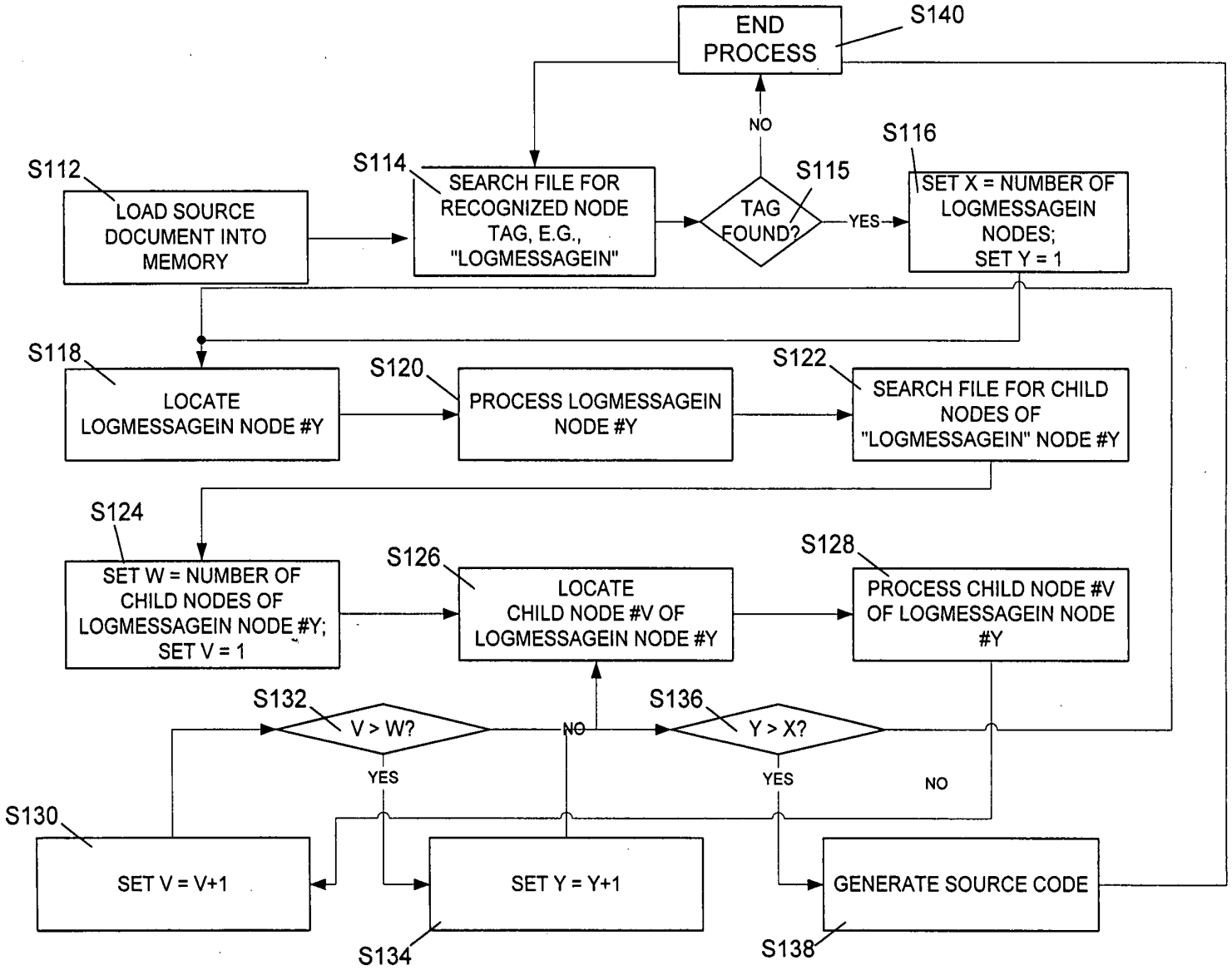


**Fig. 8**

OIPE
JUL 0 1 2005
PATENT & TRADEMARK OFFICE



```
1.      if ( !FAILED( spElement->getElementsByTagName(
2.              L"LogMessageIn", &spNodes ) )  && spNodes )
3.      {
4.              long lLength = 0;
5.              if ( FAILED( spNodes->get_length(&lLength)))
6.              {
7.                      return ;
8.              }
9.              CString sText;
10.             for ( long i = 0 ; i < lLength ;i++ )
11.             {
12.                     spNode.Release();
13.                     if ( !FAILED ( spNodes->get_item( i ,  &spNode ) ) )
14.                     {
15.                             CheckOutputLine( 0, spNode );
16.                             GetAttribute( spNode , L"id", sText );
17.                             if ( m_bDoVB )
18.                             {
19.                     sText = _T("Sub LogMessageIn_")+ sText + _T("( vtObj as VTMsgObj )");
20.                                     OutputLine( 0 , sText );
21.                                     OutputLine( 0, _T("\r\n") );
22.                                     OutputLine( 1, _T("'Temp variables used by routine\r\n"));
23.                                     OutputLine( 1, _T("Dim sTmp As String\r\n") );
24.                                     OutputLine( 1, _T("Dim sTmp4 As String\r\n") );
25.                                     OutputLine( 1, _T("Dim sTmp5 As String\r\n") );
26.                                     OutputLine( 1, _T("Dim sTmp3 As String\r\n") );
27.                                     OutputLine( 1, _T("Dim sTmp2 As String\r\n") );
28.                                     OutputLine( 1, _T("Dim sCmp As String\r\n") );
29.                                     OutputLine( 1, _T("Dim iOffset As integer\r\n") );
30.                                     OutputLine( 1, _T("\r\n"));
31.                                     OutputLine( 1, _T("On Error Goto ErrOut\r\n") );
32.                                     OutputLine( 1, _T("\r\n"));
33.                             } else
34.                             {
35.                                     AddCFunction( "LogMessageIn_" + sText);
36.                                     sText = _T("void MsgHandler::LogMessageIn_") + sText;
37.                                     sText += _T("( IDualVTMsgObj * vtObj )\r\n{\r\n\ttry {\r\n");
38.                                     OutputLine( 0 , sText );
39.                                     OutputLine( 0, _T("\r\n") );
40.                                     OutputLine( 1, _T("//Temp variables used byroutine\r\n"));
41.                                     OutputLine( 1, _T("CComBSTR sTmp;\r\n") );
42.                                     OutputLine( 1, _T("CComBSTR sTmp2;\r\n") );
43.                                     OutputLine( 1, _T("CComBSTR sTmp3;\r\n") );
44.                                     OutputLine( 1, _T("CComBSTR sTmp5;\n") );
45.                                     OutputLine( 1, _T("CComBSTR sTmp4;\n") );
46.                                     OutputLine( 1, _T("CComBSTR sCmp;\r\n") );
47.                                     OutputLine( 1, _T("int iOffset = 0, iLastPos = 0;\r\n") );
48.                                     OutputLine( 1, _T("\r\n"));
49.                                     OutputLine( 1, _T("//End of temp variables\r\n"));
50.                                     OutputLine( 1, _T("\r\n"));
51.                             }

52.                             m_bProcessingGetResponseCode += 1;
53.                             ProcessMessageIn( 1, spNode );
54                              m_bProcessingGetResponseCode  -= 1;

55.                             if ( m_bDoVB )
56.                             {
57.                                     OutputLine(0, _T("ErrOut:\r\n"));
58.                             OutputLine(1, _T("vtObj.EndSetErro Err.Number, Err.Description\r\n"));
59.                             }
```

Fig. 9

OIPE JUL 0 1 2005 PATENT & TRADEMARK OFFICE JC185



```
1. void CXMLEditDoc::ProcessMessageIn( int iTabIndex, CComPtr<IXMLDOMNode>&
spParentNode, BOOL bInBuildField, BOOL *bIfWasProccessed )
2.{
3.      CComPtr<IXMLDOMNode> spChild;
4.      if ( FAILED( spParentNode->get_firstChild( &spChild)) || !spChild )
5.      {
6.              retu rn;
7.      }

8.      CComPtr<IXMLDOMNodeList> spList = NULL;
9.      if ( !FAILED( spParentNode->get_childNodes( &spList ) )  && spList )
10.     {
11.             CComBSTR sNodeName;
12.             CComBSTR sText;
13.             CComPtr<IXMLDOMNode> spNode;
14.             long lLength = 0;
15.             spList->get_lengt  h( &lLength );
16.             for ( long i = 0 ; i < lLength ; i++ )
17.             {
18.                     spNode.Release();
19.                     sText.Empty();
20.                     sNodeName.Empty();
21.                     if ( !FAILED ( spList->get_item( i ,  &spNode ) ) )
22.                     {
23.                             spNode->get_nodeName( &sNodeName );

24.                             Che ckOutputLine( iTabIndex, spNode );

25.                             void *ptr = NULL;

26.                             CString strNodeName = sNodeName;

27.                             strNodeName.MakeUpper();

28.                             // now the ugly look up table
29.                             glb_MapOfXMLStringsTolds.Lookup( strNodeName, ptr );

30.                             int id = (int)  ptr;

31.                             switch ( id )
32.                             {
*
*
56.                             case IDTAG_BitmapDateIn:
57.                             {
58.                                     CString sBitPos;
59.                                     CString sLen;
60.                                     CString sTranField;
*
*               {
74.                                                             OutputLine( iTabIndex, _T("If
vtObj.IsBitmapPositionSet( ")  , _T("if (IsBitmapPositionSet( vtObj, "));
                                                                OutputLine( 0, sBitPos );
                                                                OutputLine( 0, _T(") Then\r\n") , _T(")) {\r\n") );
                                                                iTabIndex += 1;
                                                                // first lets get the data
                                                                CheckForPackedAttribute( iTabIndex, spNode );
```

Fig. 10

PATENT & TRADEMARK OFFICE JUL 01 2005 OIPE JC186

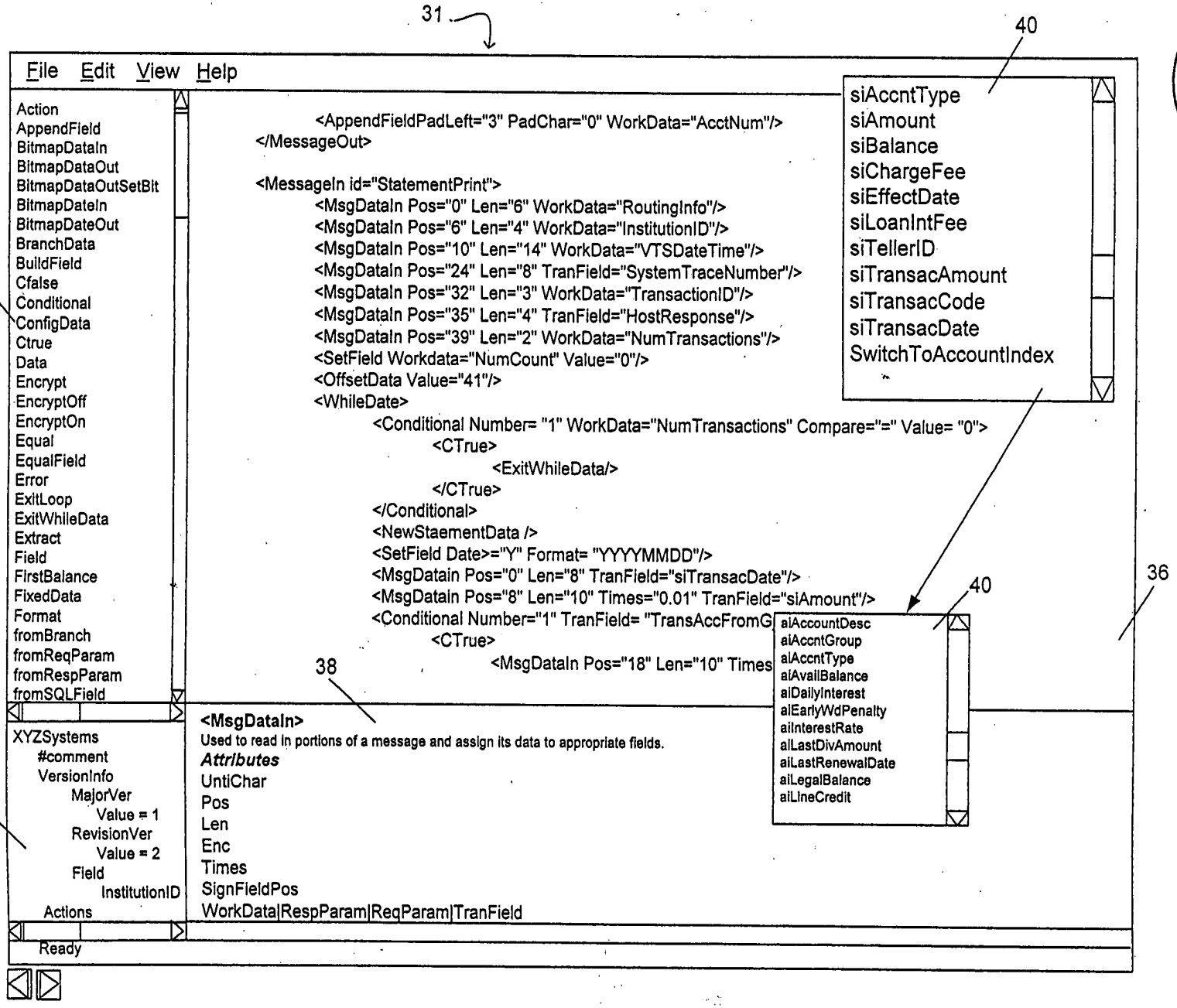


FIG. 11